



Pedro André Pereira
da Cunha

Centroproduto: Fluxo de verificação do
software



**Pedro André Pereira
da Cunha**

**Centroproduto: Fluxo de verificação do
software**

Relatório de estágio apresentado à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Matemática e Aplicações, realizada sob a orientação científica do Doutor João Pedro Antunes Ferreira da Cruz, Professor Auxiliar do Departamento de Matemática da Universidade de Aveiro.

o júri

presidente

Prof. Doutor Agostinho Miguel Mendes Agra

Professor Auxiliar da Universidade de Aveiro

vogais

Prof. Doutor João Paulo Silva Barraca

Professor Auxiliar da Universidade de Aveiro

Prof. Doutor João Pedro Antunes Ferreira da Cruz

Professor Auxiliar da Universidade de Aveiro (orientador)

palavras-chave

Centroproduto, testes, utilizador, programador, XLS, python, selenium, automação, produtos, mensagens.

resumo

Este trabalho surgiu no âmbito do estágio final do Mestrado de Matemática e Aplicações e cujo objetivo principal é apresentar todas as fases na empresa Centroproduto. São expostas sugestões de melhorias a este processo. Inicialmente foram realizados testes de software da empresa, atividade que despendeu mais tempo. Posteriormente foi desempenhada a técnica de Web Scraping, uma técnica de recolha de informações de websites para o marketing da empresa onde a descoberta da biblioteca 'Selenium' do permitiu introduzir tarefas automatizadas na empresa. Mais tarde foram elaboradas especificações do comportamento esperado, isto é, tabelas com as indicações que o grupo de testes deve proceder. Por fim, foram desenvolvidas diversas tarefas automatizadas que antes eram efetuadas manualmente.

Keywords

Centroproduto, testes, utilizador, programador, XLS, python, selenium, automação, produtos, mensagens.

Abstract

This work came under the internship of the Masters in Mathematics and Applications and the principal aim is to explain all of the tasks at the company Centroproduto.

Will be exposed all of the improvements done on the company software tests. This was the activity that make more time of the internship.

After that was developed the Web Scraping technic wich includes collect of websites information to the marketing sector at the Centroproduto.

In this tasks was very important the discover of the Selenium library, that allowed to automatized company daily tasks.

Later was created some behavic especifications which means tables with group tests indications.

At least was developed different automatized tasks that where only did manually before.

Conteúdo

Lista de Figuras	iii
Lista de Tabelas	v
1 Introdução.....	1
2 Centroproduto.....	3
2.1 Apresentação	3
2.2 Diferentes equipas e fases de trabalho.....	4
3 Teste de Software.....	7
3.1 Princípios para realizar testes de software.....	7
3.2 Fases de testes do software	8
3.2.1 Teste de unidade.....	8
3.2.2 Teste de integração.....	8
3.2.3 Teste operacional	8
3.2.4 Teste de aceitação	8
3.2.5 Teste de sistema	9
3.2.6 Teste alfa e beta.....	9
3.3 Técnicas de testes do software.....	9
3.3.1 Caixa Preta.....	10
3.3.2 Caixa Branca	11
3.3.3 Caixa Cinza	11
3.3.4 Regressão	11
3.4 Testes ao Software na empresa Centroproduto.....	12
3.5 Alterações propostas	15
3.5.1 Utilização de uma plataforma virtual como o “Lean Testing”	15
3.5.2 Modificações no XLS utilizado	16
3.5.3 Criação de um BOT	17
4 Web Scraping	19
4.1 Beautiful Soup/Requests	19
4.1.1 Beautiful Soup – Exemplo prático	21
4.2 Web Scraping na empresa Centroproduto.....	22
4.3 Alterações Propostas.....	22

4.3.1	Funções.....	23
4.3.2	Uso de uma biblioteca do Python : “Selenium”	23
5	Especificações do Comportamento Esperado	27
5.1	Especificações do Comportamento Esperado na empresa Centroproduto	27
6	Automação de tarefas.....	29
6.1	Gestão de endereços de email da empresa.....	29
6.2	Criar e Importar produtos.....	31
6.3	LINKEDIN.....	32
6.4	Descobrir o CAE de empresas.....	33
6.5	SEND MESSAGES.....	34
7	Conclusão.....	35
8	Bibliografia.....	37

Lista de Figuras

Figura 1 - Website da empresa Centroproduto	3
Figura 2 - Ciclo de instâncias da empresa	5
Figura 3 - Coluna 'Error Report'	13
Figura 5 - Coluna 'VERIFICATION'	14
Figura 4 - Coluna 'CORRETION'	14
Figura 6 - código fonte da página web 1	21
Figura 7 - Exemplo do botão 'more products' encontrado em websites.....	23
Figura 8 - código fonte da página web 2	25

Lista de Tabelas

Tabela 1 - Exemplo1 do teste da caixa preta10

Tabela 2 - Exemplo2 do teste da caixa preta10

1 Introdução

Este trabalho surgiu no âmbito do estágio final do Mestrado de Matemática e Aplicações cujo objetivo principal é apresentar todas as fases de trabalho do trabalho realizado na empresa Centroproduto [8].

O relatório é constituído por 7 capítulos, estando estes apresentados de forma a explicar todo o trabalho efetuado durante o estágio.

No próximo capítulo é apresentada a empresa Centroproduto onde o presente trabalho foi desenvolvido.

No terceiro capítulo são expostos os testes de software realizados diariamente de forma a garantir a melhor performance do website da empresa.

No quarto capítulo é exibida a técnica de Web Scraping que permite retirar informações de páginas web para poder usar na divulgação da empresa. Além deste objetivo, a técnica também é usada para criar catálogos para os utilizadores. É de salientar, que neste capítulo, para combater uma dificuldade a recolher informação de um website foi encontrada a biblioteca do Python [2], 'Selenium' [9] que permitiu, mais tarde, automatizar tarefas na empresa. Esta descoberta foi uma grande mais valia para a empresa, pois permitiu testes muito mais elaborados no website.

No quinto capítulo são apresentadas as especificações do comportamento esperado, ou seja, tabelas com as indicações que o grupo de testes deve proceder para testar uma determinada funcionalidade do website.

No sexto capítulo são expostas todas os automatismos de tarefas que foram elaboradas de forma a evitar que determinadas ações sejam repetidas, aumentando assim a eficácia e diminuindo o tempo gasto a realizar a mesma tarefa.

Por fim são apresentadas todas as conclusões gerais de todo o trabalho realizado.

2 Centroproduto

Neste capítulo é feita a descrição da empresa de onde o presente trabalho foi desenvolvido. São descritas as diferentes equipas bem com as fases de trabalho até atingir o resultado final.

2.1 Apresentação

O Centroproduto é uma empresa fundada em 2012 com sede no Campus Universitário de Santiago no Edifício 1 em Aveiro. Esta tem diversos parceiros de conhecimento e recursos humanos tais como a Universidade de Aveiro, o ISCA (Instituto Superior de Contabilidade e Administração de Aveiro), a Escola Profissional de Aveiro e o Agrupamento de Escolas de Águeda (AEAS). Além disso conta com a Incubadora de Empresas da Universidade de Aveiro (IEUA) e a Associação Industrial do Distrito de Aveiro (AIDA) como parceiros de dinamização.

O principal objetivo é proporcionar aos utilizadores um portal de negócios que utiliza algoritmos específicos para identificar os parceiros certos para cada negócio. Também promove empresas, as suas marcas e produtos, valorizando a informação completa e detalhada.

Outra das funcionalidades é agilizar a partilha de dados entre os fornecedores, revendedores e consumidor e empresa, criando novas formas de networking e interação profissional entre empresas de todo o mundo.



Figura 1 - Website da empresa Centroproduto.

2.2 Diferentes equipes e fases de trabalho

Na empresa são desempenhados vários tipos de tarefas. Normalmente estas são desenvolvidas individualmente ou por um grupo de dois elementos.

Sendo assim existe um grupo profissionais cujo principal objetivo é desenvolver a programação de software da empresa. Este grupo de programadores é dividido em quatro áreas diferentes: programadores de Front-End que abordam principalmente a interface de interação com o utilizador, programadores de Back-End que desenvolvem o sistema que vai ser utilizado para a gestão dos dados, sistema esse que terá interatividade com o utilizador e aplicará a interface que foi desenvolvida pelo Front-End. Também existe um grupo de programadores mais versáteis que tratam do webscraping (será mais tarde abordado este tema) e de programas em Python [2] para automatizar alguns processos da empresa.

Além dos programadores existe um grupo que aborda a questão do marketing com o objetivo de divulgar o máximo possível o projeto.

Visto que o website contém diversas línguas como português, inglês, espanhol, francês, etc. existe um profissional a tratar de todas as traduções.

Para assegurar que o website não contém nenhum tipo de erros existe também um grupo de testes que verifica todas combinações possíveis de ações dos utilizadores em cada menu do site. Para saber que combinações são necessárias testar também existe um grupo de profissionais que tratam de as elaborar.

A empresa possui várias instâncias, isto é, cópias do website original permitindo que a equipa de programadores e de testes possam testar as novas funcionalidades de forma a quando não existirem erros graves, poder ser possível aplicar estas mudanças no website original. Estas instâncias são:

- PREDEV;
- DEV1;
- DEV2;
- ALPHA;
- REAL.

As instâncias apenas estão disponíveis a quem estiver na rede da empresa.

A PREDEV é onde a equipa de programadores opera. Todos os testes feitos por estes é feita na PREDEV.

Após os programadores implementarem novas funções na PREDEV estas são transferidas para a DEV1 e a equipa de testes vai testar através das tabelas de testes todas as funcionalidades do website e sempre que vai encontrando algum erro regista-o num ficheiro partilhado com os programadores para que em simultâneo estes o corrijam na PREDEV.

Quando a equipa de testes termina de testar a DEV1 e a equipa de programação termina a correção destes mesmos erros, na PREDEV dá-se uma nova atualização dos dados que foram corrigidos e prontos para serem testados na DEV2.

Sendo assim volta-se a fazer os mesmos testes, desta vez na DEV2, mas tomando uma maior atenção aos erros descobertos na DEV1 de forma a garantir que estes foram corrigidos. Como foi feito anteriormente, após concluídos os testes é feita a atualização dos dados para a ALPHA.

Nesta fase, após terminar de testar todos os casos testados nas outras instâncias, caso não sejam encontrados erros graves, é feita a atualização para a REAL, website original e a equipa de programação volta para a PREDEV com o objetivo de implementar novas funcionalidades.

E este trabalho acontece de forma cíclica. Os testes na DEV1 demoram aproximadamente em média duas semanas dependendo do número de profissionais na equipa de testes.

Já os testes na DEV2 e na ALPHA são mais rápidos, demorando em média uma semana.

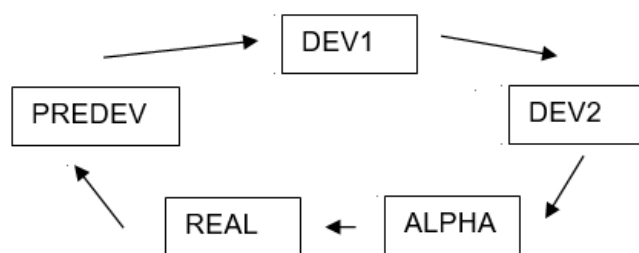


Figura 2- Ciclo de instâncias da empresa

3 Teste de Software

Os testes devem estar presentes em todas as fases de desenvolvimento do software de qualquer empresa. É muito raro existir um software sem erros. Daí esta fase ser fundamental pois assegura o bom funcionamento de todo o trabalho desenvolvido.

Para isso as empresas contratam profissionais que identificam esses erros de modo a que os programadores os possam corrigir.

Estes profissionais, até conseguirem testar todo o software, atravessam diferentes etapas e utilizam inúmeras técnicas até conseguirem chegar ao resultado final, sendo este um software com o mínimo de erros possível.

3.1 Princípios para realizar testes de software

Para realizar testes de software com a melhor qualidade possível é preciso respeitar vários princípios. O fator mais importante é seguir uma metodologia de trabalho. Este é um princípio comum em todos os tipos de trabalho. Com metodologia obtém-se sempre um trabalho mais organizado, mais completo e mais seguro.

Todos os testes devem ser efetuados com o máximo de qualidade possível. Pois o mínimo erro que poderá não ser encontrado poderá por em causa todo o trabalho realizado.

Possuir tabelas com o comportamento esperado de cada funcionalidade, é também algo essencial para realizar os testes, pois quem realiza os testes não sabe como a aplicação funciona. Assim, através destas tabelas é fácil saber se a aplicação está a funcionar como é esperado.

Os testes deverão estar focados em todas as condições possíveis e não apenas nas condições inválidas de execução. Quem realizar os testes deve ter uma **visão destrutiva** do sistema com o objetivo de encontrar erros. Este deve tentar arranjar todas as maneiras possíveis de os encontrar. Além disto é necessário utilizar recursos humanos para a implementação e verificação pois todas as pessoas são diferentes e cada pessoa toma uma determinada ação de maneira diferente. Sendo assim, será mais fácil desvendar todos os erros que possam ocorrer na aplicação.

3.2 Fases de testes do software

Para testar um software existem diferentes fases pelas quais o profissional que irá testar terá de realizar. Cada fase tem um diferente objetivo que em conjunto irão promover um teste mais desenvolvido e mais competente assegurando assim uma melhor eficácia.

3.2.1 Teste de unidade

É a fase do processo de teste onde se testam as menores unidades referentes ao software. Este processo incide em isolar pequenas partes do sistema e de seguida testá-las de forma autónoma. Este teste não garante que a integração dessas partes irá funcionar corretamente. O objetivo deste teste é de encontrar falhas de funcionamento numa determinada funcionalidade independentemente do resto do programa.

3.2.2 Teste de integração

Na fase de integração são testadas diferentes componentes combinadas de forma a averiguar se estas combinações funcionam de modo satisfatório ou se apresentam falhas provenientes da integração interna destas unidades combinadas.

Estes testes podem ser explicados como sendo vários testes de unidades realizados simultaneamente.

3.2.3 Teste operacional

O teste é realizado em ambientes de pré-produção. Esta fase é aplicável apenas a sistemas de informação próprios de uma organização, cujo acesso pode ser feito interna ou externamente. Nesta fase de teste devem ser feitas simulações para garantir que quando o sistema estiver disponível aos utilizadores, não contenha erros e seja bem sucedida.

3.2.4 Teste de aceitação

Os testes de aceitação são realizados por um grupo restrito de utilizadores do sistema a testar, que simulam operações de rotina de modo a verificar se o

comportamento está de acordo com o solicitado. É um teste formal conduzido para determinar se um sistema satisfaz ou não os seus critérios de aceitação e para permitir ao cliente determinar se aceita ou não o sistema.

3.2.5 Teste de sistema

Nesta fase de teste o objetivo principal é executar o sistema do ponto de vista do utilizador final, testando todas as funcionalidades da aplicação em busca de falhas em relação aos objetivos originais. Os testes são executados em condições similares de ambiente, ou seja, àquelas que o utilizador final irá utilizar.

3.2.6 Teste alfa e beta

Os testes requerem fases antes do produto ser disponibilizado a todos os usuários. Sendo assim, entre a finalização da aplicação e a entrega é denominada a fase alfa. Nesta altura são realizados os testes alfa, onde o grupo restrito de usuários utiliza a aplicação de modo a detetar possíveis falhas.

Completada esta fase, será lançada também a um grupo restrito de utilizadores versões de teste do sistema denominadas fase beta. Esta também é um teste de aceitação onde os utilizadores que a irão testar normalmente são da empresa que vai usufruir do sistema. Nesta fase o programador não está presente.

Sendo assim o teste será realizado num ambiente que não poderá ser controlado pelo mesmo. Após a realização do teste, os utilizadores relatam os problemas que encontraram durante esta fase e os programadores fazem a modificação para depois poder partilhar com todos os utilizadores.

3.3 Técnicas de testes do software

Existem algumas técnicas que são uma mais valia cujo objetivo continua a ser o encontrar falhas no sistema. Estas serão descritas de seguida.

3.3.1 Caixa Preta

Também chamado de teste funcional, este teste avalia o comportamento externo do software sem se considerar o comportamento interno do mesmo. Este teste consiste principalmente em usar os dados de entrada fornecidos anteriormente no teste e comparar os resultados obtidos com o comportamento esperado.

Quanto mais entradas forem fornecidas mais rico será o teste. Sendo assim, a situação ideal seria testar todas as entradas possíveis, mas como é lógico isso seria, na maioria dos casos, impossível. Posto isto a melhor abordagem para aplicar esta técnica é escolher um subconjunto de entradas que maximize a riqueza do teste. Um exemplo que facilmente explica este teste é o seguinte:

Entrada	Caracteres Permitidos	Casos de Teste
Nome do Produto	Entre 2 e 255	7 caracteres
Características	Entre 1 e 5000	2000 caracteres
Referencia	Entre 1 e 18	10 caracteres

Tabela 1 – Exemplo1 do teste da caixa preta

Além disto também é feita uma análise de valor limite, ou seja, os valores que se encontram nas fronteiras do domínio de entrada são testados. Para o caso anterior teríamos:

Entrada	Caracteres Permitidos	Casos de Teste
Nome do Produto	Entre 2 e 255	1 caracter
		2 caracteres
		255 caracteres
		256 caracteres
Características	Entre 1 e 5000	0 caracteres
		1 caracter
		5000 caracteres
		5001 caracteres
Referência	Entre 1 e 18	0 caracteres
		1 caracter
		18 caracteres
		19 caracteres

Tabela 2 - Exemplo2 do teste da caixa preta

Assim é possível maximizar a probabilidade de detetar erros. Esta técnica é aplicada a todas as fases de teste.

3.3.2 Caixa Branca

Sendo um teste mais orientado à lógica, a técnica da caixa branca baseia-se em avaliar o comportamento interno do software. Esta técnica recai diretamente no código do programa analisando o código do programa e elaborando tabelas de teste que cubram todas as possibilidades do software.

A maior vantagem deste teste é que como a estrutura interna é usada como referência, é fácil encontrar os valores de entrada mais úteis para o teste o que ajuda na otimização geral do sistema.

Esta técnica é recomendada para as fases de Teste de Unidade e de Integração, cuja responsabilidade principal fica a cargo dos programadores visto conhecerem melhor o código produzido.

3.3.3 Caixa Cinza

A técnica da caixa cinza é uma mistura das técnicas: caixa branca e caixa preta. Isto é o profissional que irá fazer os testes não tem acesso ao código fonte mas sim tem conhecimento dos algoritmos que foram implementados.

3.3.4 Regressão

É uma técnica aplicável a novas versões do software ou á necessidade de executar um novo ciclo de teste durante o processo de desenvolvimento. Sendo assim após cada nova versão do software ou a cada ciclo são efetuados os mesmos testes de forma a assegurar que os erros encontrados anteriormente foram corrigidos.

3.4 Testes ao Software na empresa Centroproduto

Testar o software é umas das tarefas mais importantes da empresa. Um website com falhas graves é o principal fator para o utilizador deixar de usar o nosso projeto. Normalmente estes testes requerem um grande número de recursos humanos, estando entre 2 a 5 pessoas a desempenhar esta tarefa.

Para começar a testar uma determinada funcionalidade do software é necessário possuir tabelas sobre o comportamento esperado do mesmo. Estas tabelas possuem a ação que devemos efetuar, o browser, o dispositivo (desktop ou mobile) bem como o resultado esperado. Estas serão totalmente explicadas no capítulo 5.

Como cada pessoa tem uma forma diferente de explicar o mesmo acontecimento, por vezes os programadores não entendem o erro encontrado, ou então não entendem todos os passos seguidos até encontrar o mesmo. Sendo assim após encontrar um erro, a equipa de testes através da aplicação Gifcam volta a reproduzir o mesmo erro para poder gravá-lo.

A aplicação Gifcam é um programa gratuito para Windows que permite criar gifs a partir da captura da tela do computador [7].

Para anotar os erros encontrados o grupo de testes e o grupo de programadores possuem um ficheiro em comum. Este ficheiro está dividido em três partes:

- 'Error report' onde os profissionais de teste ao encontrar as falhas preenchem (figura 3);
- 'Correction' onde os programadores após corrigir os erros partilham a informação (figura 4);
- 'Verification' onde os profissionais que estão a testar o software verificam se o erro foi corrigido e indicam a conclusão. (figura 5)

ERROR REPORT									
ID	Date	Author	Browser	LOGIN	LA	URL	Error type	Symptom/Suggestions	Comment
2495									
2494									
2493									
2492	20170407	SLD	Qualquer	ANT		https://dmsa. Mobile	SUGESTÃO Catalog view	Quando uma empresa não tem catálogo, a dropdown da língua dos conteúdos não faz qualquer ação, apesar de ter a setinha a indicar para abrir.	SLD: Provavelmente, para estes casos, o melhor seria a dropdown nem aparecer ou ficar como no desktop onde não tem a setinha para carregar.
2491	20170407	PPC	Qualquer	ANT		https://dmsa. BOTH Desktop	BAIXO Landing	Após efetuar uma pesquisa na uma inconsistência entre mobile e desktop. No desktop o campo procurado continua na landing enquanto que no mobile este campo após ser pesquisado na primeira vez desaparece da landing.	
2490	20170407	PPC	Qualquer	NOVO		https://dmsa. Desktop	ALTO Catalogo	Se enviar um invite através da página do catálogo e no fim do invite voltar a carregar onde diz "novo invite" isto volta a criar os campos automaticamente.	
2489	20170407	PPC	Qualquer	ANT		https://dmsa. BOTH	BAIXO Edit_Profile	Quando tenho um nif já existente e mudo o País para Cuba, por exemplo, é necessário fazer uma alteração no NIF como por exemplo dar um espaço para poder validar. No tablet além disto não autoriza que crie um NIF num País que já tinha sido usado noitro.	
2488	20170407	DP	Qualquer	NOVO		https://dmsa. Mobile	ALTO Pesquisa	Com telemovel em landscape, ao fazer pesquisa por produtos, com resultados de empresas que tenham o CAE comprado, o nome da empresa sobe e fica cortado e sobreposto, necessario alinhar detalhes para ver o erro.	
2487	20170407	SLD	Qualquer	ANT		https://dmsa. Desktop	SUGESTÃO Pre-actor	Depois de reclamar um pre-actor, o ecrã da SUP não é atualizado e a empresa continua na lista (apesar de já não ser pre-actor).	IDP - do que me lembro sempre foi assim, não sei qual seria o CE ESO - está programado para que assim seja, existe até um campo nessa tabela "SEN" TO ACTOR DATEIME" que indica o dia e hora em que esse pre-actor se tomou actor.
2486	20170407	PPC	Qualquer	ANT		https://dmsa. Desktop	BAIXO Edit_Profile	Se o utilizador não tiver foto de perfil e carregar no "enter" em qualquer um dos campos, vai abrir uma barra a perguntar a localização da imagem para colocar na foto de perfil.	
2485	20170407	DP	Qualquer	NOVO		https://dmsa. Mobile	ALTO Pesquisa	Quando a unidade comprida de um produto com preço, e vamos a pesquisa do produto (lista), a unidade sai roa da area normal, ainda manda para baixo o país da empresa do produto.	
2484	20170407	PPC	Qualquer	ANT		BOTH	SUGESTÃO Mensagens	Ao responder a uma mensagem se carregar na landing aparece um pop up a perguntar se realmente queremos sair da mensagem. Se em vez disso criarmos uma mensagem e fizermos o mesmo procedimento o pop up já não aparece e acho que deveria aparecer.	
2483	20170407	DP	FF	ANT		https://dmsa. Desktop	BAIXO Image Control	Ao passar o cursor por cima das imagens, aparece um quadrado, so o contorno (ver gif)	
2482	20170407	SLD	Qualquer	ANT		https://dmsa. BOTH	MEDIO Pre-actor	Nas empresas similares, se estiver em modo de Ver detalhes, ao mudar de página os detalhes não ficam visíveis. Tenho de passar o rato por cima dos logos ou voltar a clicar no botão de Ver detalhes para aparecerem.	
2481	20170407	SLD	Qualquer	ANT		https://dmsa. BOTH	ALTO Pre-actor	Nas empresas similares por CAE, empresas que tenham um CAE que harmoniza com o pre-actor mas que tenham um país diferente do pre-actor não estão a aparecer.	Dados usados: Pre-actor: País CUBA // CAE 4510 Actor 1: País CUBA // CAE 4510 Actor 2: País PORTUGAL // CAE 49120

Figura 3 - Coluna 'Error Report'

CORRECTION			
In Charge	STATUS	Correction to Implement	n° / CAL/DEV/ALPHA/R
FE			
MSO			
MSO	CORRIGIDO		LOCAL
MSO			
SSC	CORRIGIDO		LOCAL
ESO			
MSO			
SSC	CORRIGIDO		LOCAL
MSO			
SSC			
MSO			

Figura 5 – Coluna ‘CORRECTION’

VERIFICATION			
In Charge	STATUS	Comments	
IDP	VERIFICADO	Comportamento inconsistente, ja foi revisto	

Figura 4 – Coluna ‘VERIFICATION’

Cada linha do ficheiro corresponde a um erro. Estas linhas possuem quatro cores possíveis:

- **Branca:** significa apenas uma sugestão.
- **Verde:** assinala um erro ligeiro que não impede o utilizador de fazer uma determinada ação. Um erro ortográfico é um exemplo de um erro verde.
- **Amarela:** indica um erro grave. Ou seja, também não impede o utilizador de efetuar uma ação, mas tem um carácter mais grave. Por exemplo, um botão desaparece e é necessário atualizar a página para este aparecer é um exemplo de um erro amarelo.
- **Vermelha:** designa um erro muito grave. Este tipo de erros impossibilitam o utilizador de desempenhar várias ações. Por exemplo, não ser possível enviar uma mensagem no nosso website é um erro muito grave. Os programadores têm de corrigir obrigatoriamente este tipo de erros.

Visto que todos os erros devem ser corrigidos é por vezes impossível para os programadores terem tempo para os eliminarem, pois estes estão constantemente a programar novas funcionalidades para o software. Sendo assim os erros têm de ser corrigidos de acordo com o nível de gravidade e os erros vermelhos têm de ser obrigatoriamente corrigidos.

A coluna de '**Error report**' tem as seguintes sub colunas:

- **ID**: permite atribuir um número ao erro.
- **Date**: data em que o erro foi encontrado.
- **Author**: quem descobriu o erro.
- **Browser**: o browser onde o erro foi encontrado: chrome, mozilla, opera ou internet explorer.
- **Login**: Se o erro é novo ou antigo.
- **URL**: URL onde colocámos o GIF/imagem do erro.
- **Device**: Se o erro foi encontrado no desktop ou no mobile.
- **Error Type**: Se o erro é leve, grave, muito grave ou apenas uma sugestão. Ao escolher uma destas opções do menú dropdown, a linha automaticamente muda de cor.
- **Symptom**: Onde descrevemos o erro.
- **Comment**: Algum comentário extra que possa ajudar na correção. Esta informação não é de carácter obrigatório.

As colunas '**Corretion**' e '**Verification**' são semelhantes. Sendo assim tem as seguintes subcolunas:

- **In charge**: O programador que deve corrigir/verificar o erro.
- **Status**: Indica se o erro já foi corrigido/verificado.
- **Comment**: Algum comentário sobre a correção/verificação do erro. Não é de carácter obrigatório.

3.5 Alterações propostas

No decorrer do estágio foram evidenciadas numerosas anomalias que poderiam ser modificadas de forma a melhorar a performance rentabilizando o trabalho e assegurando assim uma melhor performance. Sendo assim foram recomendadas as seguintes alterações expostas de seguida:

3.5.1 Utilização de uma plataforma virtual como o “Lean Testing”

Na empresa são utilizados diversos ficheiros Excel contendo a informação dos testes que devemos realizar, bem como com os erros encontrados, como o comportamento esperado para cada um dos testes.

Em alternativa poderia ser usado uma plataforma como o Lean Testing. Esta plataforma tem diversas funcionalidades [1]:

- estatística acerca dos erros obtidos informando o tipo de erro e a sua data;
- interface limpa;
- notificações dos erros.

O Lean Testing tem as seguintes vantagens e desvantagens:

Vantagens:

- boa organização;
- gráficos e estatísticas sobre os erros obtidos;
- facilidade em verificar se um determinado erro já foi encontrado.

Desvantagens:

- para implementar esta plataforma seria necessário bastante tempo para introduzir todos os dados referentes às tabelas do comportamento esperado na aplicação;
- caso seja necessário voltar a usar o XLS será bastante difícil recuperar os dados da plataforma;
- é uma aplicação paga;
- a aplicação fica com muita informação interna sobre a empresa.

Esta proposta não foi aceite pela empresa devido às desvantagens evidenciadas anteriormente.

3.5.2 Modificações no XLS utilizado

Como não foi possível a utilização de uma plataforma como a sugerida anteriormente foi proposta a adaptação do ficheiro XLS que utilizamos para algo mais próximo desta mesma plataforma:

- **Remover informação duplicada:** em algumas ações tínhamos caminhos que já tinham sido feitos anteriormente.
- **Criação de uma coluna com o ecrã do erro:** para uma busca mais clara sobre se um erro já foi encontrado ou não.
- **Criação de gráficos:** para uma melhor perceção do número de erros, a sua classificação e o ecrã em que este erro foi descoberto.
- **Alteração da ordem das rotas.**
- **Adicionar GIFS ou imagens às rotas:** para uma informação mais clara e concisa.

Das sugestões acima propostas apenas não foi concretizável adicionar GIFS ou imagens às rotas do comportamento esperado pois seria necessário despende muito tempo.

3.5.3 Criação de um BOT

Para uma diminuição do tempo dispensado dos recursos humanos responsáveis por testar o software da empresa a criar as condições necessárias e para assegurar uma maior eficácia seria útil a utilização de um BOT.

Este poderia ter duas utilizações.

- Testar;
- Criação de dados para os testes.

No que diz respeito a testar o software não seria muito útil visto que as funcionalidades do website estão constantemente a ser modificadas e levaria a uma enorme sobrecarga na área da programação e não seria bom ao nível do custo/benefício.

Caso as rotas fossem constantes ao longo do tempo aí sim seria uma enorme mais valia.

Em relação à criação de dados seria bastante benéfico pois existem variadas rotas onde é gasto demasiado tempo apenas na criação dos dados que serão utilizados.

Sendo assim foi criado um programa na linguagem Python com bastantes funcionalidades.

Este programa tem como objetivo principal a criação de utilizadores e empresas numa funcionalidade da nossa plataforma denominada de 'fast insert' de forma imediata. O profissional que irá usar o programa apenas tem de preencher um XLS, que se encontra na mesma pasta que o programa.

Cada linha do XLS corresponde a um utilizador e empresa diferente. As colunas evidenciadas de seguida a negrito indicam que possuem um carácter obrigatório. O XLS possui as seguintes colunas:

- User Status: O utilizador do programa tem um menu dropdown nesta coluna com a opção 'READY'. Se esta célula estiver vazia o programa irá passar ao utilizador seguinte. Caso contrário, e esta célula tiver a opção 'READY' escolhida, o programa criado irá ler a linha.
- Program Status: Quando o programa acaba de ler a linha, este cria o utilizador e a empresa, e automaticamente preenche a célula com 'DONE'.
- **User email:** Email do utilizador.
- User password: Ao criar o utilizador e empresa do mesmo, o website automaticamente atribui uma password a este que só os administradores têm acesso. O programa faz login na conta do administrador e após ir ao ecrã onde a password se encontra recolhe e automaticamente guarda-a nesta célula do XLS.
- User fname: Primeiro nome do utilizador.

- User Iname: Último nome do utilizador.
- User Lang: Língua do mesmo.
- User Country: País deste.
- User Phone number: Número de telefone do utilizador.
- **Actor Legalname:** Nome legal da empresa.
- **Actor Shortname:** Nome da empresa.
- **Actor NIF:** NIF da empresa.
- Actor URL: URL da empresa.
- Actor CAE: Código da atividade económica de empresa.
- **Actor Email:** Email da empresa.
- Actor Phone number: Número de telefone da empresa.
- Actor Street: Rua onde a empresa está alojada.
- Actor Floor: Andar onde a empresa se encontra.
- Actor PostalCode: Código postal da empresa.
- Actor Parish: Freguesia onde a empresa está acomodada.
- Actor County: Concelho onde a empresa se encontra.
- **Actor Country:** País da empresa.
- Actor District: Distrito onde a empresa está alojada.
- Actor Facebook: Facebook da empresa.
- Actor Linkedin: Linkedin da empresa.
- Actor Twitter: Twitter da empresa.
- Actor Youtube: Youtube da empresa.
- Actor Description: Descrição da empresa.
- Actor Advantages: Vantagens da empresa.

Caso alguma esteja em branco o programa irá perguntar ao utilizador se deseja preencher a mesma célula ou se prefere passar para o próximo utilizador.

Após o utilizador do programa preencher todo o XLS, o programa irá criar todos os utilizadores e a sua respetiva empresa com todos os dados inseridos de forma bastante mais rápida do que se fosse feito manualmente.

4 Web Scraping

Atualmente, é cada vez maior, a importância da recolha de dados de websites. Estes são escritos em linguagem HTML, ou seja, cada página é um documento com estrutura nesse tipo de linguagem[5].

A técnica de Web Scraping, como o nome indica, é uma técnica que permite que um programa consiga retirar todos os dados que lhe interessam de uma página web, mantendo uma certa estrutura, isto é, sem que estes sejam alterados.

O principal objetivo desta é que o utilizador consiga através de um só programa, examinar várias páginas da web, reunir e recolher os dados que necessita, preservando a estrutura dos mesmos, na forma que lhe for mais conveniente.

Por vezes, quando os dados que se desejam extrair são reduzidos é favorável que se extraiam manualmente com o uso do “copy past” habitual. Quando isso não acontece, o utilizador para fazer a recolha iria demorar bastante tempo e somente nesses casos é que se usa o Web Scraping.

Um bom exemplo de Web Scraping é pensar num website a simular as páginas amarelas onde contém o número de telefone de todas as empresas de Portugal e pretender recolher esses mesmos números de telefone. Manualmente seria uma tarefa que iria demorar bastante tempo para concretizar. Usando um programa rapidamente se recolhe toda a informação.

4.1 Beautiful Soup/Requests

A Beautiful Soup é uma biblioteca do Python usada para Web Scraping. Esta é usada em parceria com a biblioteca Requests.

Enquanto que a biblioteca Requests é usada para transferir o conteúdo HTML da página desejada para uma variável, a biblioteca Beautiful Soup procura a informação desejada nessa variável.

Para instalar as bibliotecas é necessário abrir a linha de comandos do sistema operativo e digitar:

```
$ pip install bs4  
$ pip install requests
```

Para estas poderem ser usadas no programa será necessário importá-las através do código:

```
> from bs4 import beautiful soup
> import requests
```

Para fazer download da página web é usado o método get da biblioteca requests.

Posteriormente, apenas é necessário guardar os dados no formato desejado usando a BeautifulSoup com o comando:

```
> page = requests.get(link desejado)
> BeautifulSoup(page.content,'html.parser')
```

De forma a recolher a informação, é necessário inspecionar a página onde desejamos recolher a mesma com a ajuda da tecla “f12” e descobrir onde é que a informação se encontra.

Posto isto usa-se o método “find” da biblioteca BeautifulSoup para extrair informação desejada.

```
>find(class_ = 'classname')
>find(id = 'idname')
>find(style = 'stylename')
>find('a')
>find('p')
>...
```

Caso a informação esteja toda no mesmo local, ou seja na mesma class, paragrafo, etc. é usado o método “find_all” onde irá retornar uma lista com toda a informação:

```
>find_all(class_ = 'classname')
>find_all(id = 'idname')
>find_all(style = 'stylename')
>find_all('a')
>find_all('p')
>...
```

Também é possível baixar imagens ou pdfs do url. Bastando para isso aplicar o seguinte código:

```
>image = requests.get(imagelink).content
>with open("%s" %imagenname, "wb") as j:
>    j.write(image)
```

4.1.1 Beautiful Soup – Exemplo prático

Na área da programação a melhor maneira de perceber o uso de uma determinada biblioteca é através de exemplos. Sendo assim de seguida irei demonstrar como é que através desta biblioteca se pode recolher informação de um website aleatório, neste caso: “<https://diretorio.info/>”.

1. Abrir link e escolher uma empresa (Exemplo: ‘aquafixe-parque-aquatico’)
2. Importar bibliotecas e fazer download da página no formato desejado (html)

```
>import requests
>from bs4 import BeautifulSoup
>link = 'https://diretorio.info/item/aquafixe-parque-aquatico/'
>page = requests.get(link)
>soup = BeautifulSoup(page.content,'html.parser')
```

3. Inspeccionar a informação que desejamos extrair usando F12 na página web

```
▼<div class="item-content-wrap" itemscope itemtype="http://schema.org/LocalBusiness">
  <meta itemprop="name" content="Aquafixe &#8211; Parque aquático">
  <meta itemprop="image" content="https://diretorio.info/wp-content/uploads/aque-fixe.png">
  <meta itemprop="address" content="Cabanas de Baixo - Foz do Sabor, Rua Lameirão, 5160-033 Torre de Moncorvo">
  ▶<div class="column-grid column-grid-3">...</div>
  ▼<div class="column-grid column-grid-2">
    ▼<div class="column column-span-2 column-narrow column-last">
      ▼<div class="address-container">
        ▶<h2>...</h2>
        ▼<div class="content">
          ▼<div class="address-row row-postal-address" itemscope itemtype="http://schema.org/PostalAddress">
            ▶<div class="address-name">...</div>
            ▼<div class="address-data" itemprop="streetAddress">
              <p>Cabanas de Baixo - Foz do Sabor, Rua Lameirão, 5160-033 Torre de Moncorvo</p>
            </div>
          </div>
          ▼<div class="address-row row-gps" itemscope itemtype="http://schema.org/Place">
            ▼<div class="address-name">
              <h5>GPS:</h5>
            </div>
            ▼<div class="address-data" itemprop="geo" itemscope itemtype="http://schema.org/GeoCoordinates">
              ▶<p>...</p>
            </div>
          </div>
          ▶<div class="address-row row-telephone">...</div>
          ▼<div class="address-row row-email">
            ▼<div class="address-name">
              <h5>E-mail:</h5>
            </div>
            ▼<div class="address-data">
              ▶<p>
                <a href="mailto:info@aquafixe.com" target="_top" itemprop="email">info@aquafixe.com</a>
              </p>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
```

Figura 6 - código fonte da página web 1

4. Recolher a informação

```
>nome = soup.find('h1').text
>endereco = soup.find(class_='address-data').text
>telefone = soup.find(itemprop='telephone').text
>email = soup.find(itemprop='email').text
>url = soup.find(itemprop='url').text
```

5. Recolher imagem da empresa

```
>logo_aux = soup.find(class_='cat-icon').find('img')['src']  
>logo = requests.get(logo_aux).content  
>with open("%s" %'logoempresa.png', 'wb') as j:  
>    j.write(logo)
```

4.2 Web Scraping na empresa Centroproduto

Na empresa Centroproduto esta técnica é usada frequentemente na área de marketing. Previamente são realizadas pesquisas de websites com dados referentes a empresas, obtendo principalmente a sua categoria empresarial e o seu endereço de email.

Posto isto, é feito um pequeno estudo do website de forma a identificar a melhor técnica de web scraping a ser usada, bem como a melhor maneira de recolher os dados.

Por fim apenas é necessário efetuar o código com as instruções expostas anteriormente e guardar os dados num ficheiro em XLS. Estes dados são guardados na base de dados e mais tarde serão usados para enviar emails a divulgar a empresa.

Além desta tarefa, a empresa Centroproduto, para facilitar a tarefa dos clientes, oferece-se para criar o perfil das empresas destes bem como os seus catálogos. Mas, como algumas empresas têm demasiados produtos, esta tarefa iria despende demasiado tempo. Sendo assim também é utilizada a técnica de Web Scraping que nos permite efetuar o mesmo trabalho usando o mínimo tempo possível.

A empresa Centroproduto para desempenhar esta acção usava a biblioteca do python 'Beautiful Soup' .

4.3 Alterações Propostas

Ao longo do percurso do estágio nesta área foram evidenciados diversos aspetos que poderiam ser melhorados de forma a rentabilizar o tempo de execução das tarefas da empresa Centroproduto que serão demonstrados de seguida.

4.3.1 Funções

O uso de funções nos programas foi o primeiro aspeto que foram cruciais. Esta implementação tem várias vantagens como:

- Clareza no código;
- Código mais reduzido;
- Facilidade em corrigir o código;
- Fácil implementação;

Sendo assim só existem vantagens no uso de funções e então estas começaram a estar presentes em todos os códigos usados na empresa.

4.3.2 Uso de uma biblioteca do Python : “Selenium”

Outro dos problemas encontrados foi o facto de ao importar produtos de uma determinada empresa esta não ter todos os produtos visíveis no seu website e em vez disso possuir um botão “More products” que apenas mostra todos os produtos ao clicar no botão, algo que usando a biblioteca “Beautiful Soup” não seria possível extrair os produtos pois esta biblioteca não permite seleccionar botões.

Para resolver este problema foi encontrada a biblioteca do Python Selenium, que simula as acções do utilizador numa página web, como cliques, scroll down, etc...



Figura 7 - Exemplo do botão 'more products' encontrado em websites

Para instalar a biblioteca é necessário abrir a linha de comandos do sistema operativo e digitar:

```
$pip install selenium
```

Para esta poder ser usada num programa Python será necessário importá-la através do código:

```
>from selenium import webdriver
```

Semelhante às bibliotecas Beautifull Soup e ao Requests para fazer download da página web é usado o método get.

Sendo assim o comando utilizado é:

```
> driver = webdriver.Navegador_que_deseja_usar()
```

```
>driver.get("link")
```

Este comando irá abrir o link num separador anónimo.

Para descobrir em que instância se encontra a informação que desejamos recolher também temos de inspecionar o elemento, usando a tecla 'f12'.

Posto isto apenas temos de usar os seguintes comandos de acordo com a instância desejada[4]:

```
>find_element_by_id("idname")
>find_element_by_class_name("classname")
>find_element_by_name("name")
>find_element_by_tag_name("tag")
>find_element_by_xpath("xpath")
>find_element_by_css_selector("css")
>...
```

Esta técnica foi um passo importantíssimo na empresa por várias razões:

- Combate os problemas evidenciados anteriormente.
- Permite testes muito mais eficientes: como biblioteca que simula ações em páginas web então esta foi utilizada para criar programas de automatização de tarefas na empresa que serão abordadas no capítulo 6.

4.3.2.1 Selenium: Exemplo prático

Para explicar melhor esta biblioteca será mostrado um exemplo onde será recolhida informação sobre uma empresa inscrita em: <https://diretorio.info/> , como no exemplo anterior (Beautiful Soup).

1. Abrir link e escolher uma empresa. (Exemplo: 'aquafixe-parque-aquatico')
2. Importar bibliotecas e abrir link

```
>from selenium import webdriver
>link = 'https://diretorio.info/item/aquafixe-parque-aquatico/'
>driver = webdriver.Chrome()
>driver.get(link)
```

3. Inspeccionar a informação que desejamos extrair

```
▼<div class="item-content-wrap" itemscope itemtype="http://schema.org/LocalBusiness">
  <meta itemprop="name" content="Aquafixe &#8211; Parque aquático">
  <meta itemprop="image" content="https://diretorio.info/wp-content/uploads/aque-fixe.png">
  <meta itemprop="address" content="Cabanas de Baixo - Foz do Sabor, Rua Lameirão, 5160-033 Torre de Moncorvo">
  ▶<div class="column-grid column-grid-3">...</div>
  ▼<div class="column-grid column-grid-2">
    ▼<div class="column column-span-2 column-narrow column-last">
      ▼<div class="address-container">
        ▶<h2>...</h2>
        ▼<div class="content">
          ▼<div class="address-row row-postal-address" itemscope itemtype="http://schema.org/PostalAddress">
            ▶<div class="address-name">...</div>
            ▼<div class="address-data" itemprop="streetAddress">
              <p>Cabanas de Baixo - Foz do Sabor, Rua Lameirão, 5160-033 Torre de Moncorvo</p>
            </div>
          </div>
          ▼<div class="address-row row-gps" itemscope itemtype="http://schema.org/Place">
            ▼<div class="address-name">
              <h5>GPS:</h5>
            </div>
            ▼<div class="address-data" itemprop="geo" itemscope itemtype="http://schema.org/GeoCoordinates">
              ▶<p>...</p>
            </div>
          </div>
          ▶<div class="address-row row-telephone">...</div>
          ▼<div class="address-row row-email">
            ▼<div class="address-name">
              <h5>E-mail:</h5>
            </div>
            ▼<div class="address-data">
              ▼<p>
                <a href="mailto:info@aquafixe.com" target="_top" itemprop="email">info@aquafixe.com</a>
              </p>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
```

Figura 8 - código fonte da página web 2

4. Recolher a informação

```
>nome = driver.find_element_by_tag_name('h1').text
>endereco = driver.find_element_by_class_name("address-data").text
>telefone = driver.find_element_by_xpath("//span[@itemprop='telephone']").text
>email = driver.find_element_by_xpath("//a[@itemprop='email']").text
>url = driver.find_element_by_xpath("//a[@itemprop='url']").text
>print(nome + "\n" + endereco + "\n" + telefone + "\n" + email + "\n" + url)
>driver.close()
```


5 Especificações do Comportamento Esperado

Para ajudar a equipa de programação na realização de novas funcionalidades para o website é costume a realização de tabelas em que se demonstra o comportamento que tal funcionalidade deverá ter. Estas tabelas além de exibir todos os caminhos possíveis até chegar à página de uma determinada funcionalidade, estas tabelas, também permitem à equipa de testes um trabalho mais eficaz, pois possuem todas as ações que os utilizadores poderão tomar. Sendo assim será necessário testar as ações indicadas no XLS.

5.1 Especificações do Comportamento Esperado na empresa Centroproduto

Todas as funcionalidades do website possuem um arquivo XLS com o comportamento esperado de cada ação. Quando a equipa de programação decide implementar a nova funcionalidade é sempre necessária a elaboração do comportamento esperado dessa funcionalidade para esta equipa saber o que deverá acontecer nesse ecrã, em qualquer das ações que o utilizador final efetuará.

A tarefa desempenhada nesta empresa foi a elaboração da tabela do comportamento esperado da funcionalidade “Import XLS” que se explica de seguida.

O website da empresa Centroproduto de forma a automatizar a tarefa de importar produtos/linhas de catálogo para a sua empresa, criou um ecrã onde o utilizador final apenas terá de descarregar o XLS correspondente à categoria dos seus produtos (caso ainda não o possua) e preenchê-lo corretamente. Posto isso apenas tem de o submeter neste mesmo ecrã. Esta funcionalidade permite três casos possíveis: importar produtos, importar linhas de catálogo e ambas, isto é, importar produtos e introduzi-los nas linhas de catálogo.

Esta simples ação requer bastantes testes até poder ser utilizada.

O XLS possui enumeras folhas.

A primeira folha, ‘DOC REVISION’ serve sempre para indicar o nome da funcionalidade, a pessoa que o criou e a data.

A folha seguinte, ‘DESIGN’ apresenta o ecrã da funcionalidade bem como a estrutura do XLS que o utilizador irá descarregar e preencher.

De seguida existe uma folha que possui um índice, denominado 'BEHAVIOUR'.

Esta tabela possui mais 12 folhas, estas destinadas aos testes:

- **'Access and Permissions'**: Destina-se a inserir o link relativo ao ecrã que iremos testar, usando todas as combinações possíveis e verificar se acontece o que é desejado.
- **'Import xls file sheet's names'**: O objetivo é combinar as diferentes flags do ecrã e fazer download do XLS e verificar se este possui as sheets corretas.
- **'Import page default contents'**: Verificar se o ecrã principal da funcionalidade está correto.
- **'Validations'**: Tem como finalidade testar todas as validações do XLS.
- **'Report – Item'**: Verificar todas as mensagens de erro da sheet 'ITEM' do XLS.
- **'Report – File to'**: Certificar que todas as mensagens de erro da sheet 'FILE TO' do XLS.
- **'Report – Catalog'**: Confirmar as mensagens de erro da sheet 'CATALOG do XLS.
- **'Report – Item Catalog'**: Apurar se todas as mensagens de erro da sheet 'ITEM CATALOG' do XLS se encontram corretas.
- **'File upload'**: Quando o XLS está preenchido é preciso importá-lo no ecrã da funcionalidade. Esta tabela certifica-se se o ficheiro é importado sem erros.
- **'Final upload'**: Verifica se os dados são importados após ser feito o upload do XLS.
- **'Result Final'**: Assegurar se aparecem mensagem de erro quando se importam os dados.
- **'Import XLS by minibank'**: Semelhante à tabela anterior, mas com o upload do XLS a ser efetuado num ecrã diferente.

As folhas anteriores possuem todos os testes que são necessários fazer para garantir que a nova funcionalidade está a operar corretamente. Em cada folha está presente o ID da ação, o dispositivo (mobile, desktop ou ambos) que indica onde deverá ser efetuado o teste e a 'start condition', ou seja os pré-requisitos para efetuar a ação.

Além destas colunas as folhas do XLS têm a ação que deverá ser efetuada, o que deverá acontecer após efetuar a ação e uma coluna para indicar se ocorreu algum erro ou se está tudo a funcionar corretamente.

Esta funcionalidade possui 912 ações diferentes.

6 Automação de tarefas

Com a introdução da biblioteca 'Selenium' do Python na empresa foi possível realizar tarefas que antes não tinham sido efetuadas.

Existem inúmeras vantagens em automatizar tarefas como:

- **Evitar tarefas repetidas:** Quando um profissional está sempre a efetuar a mesma ação é comum este efetuar erros.
- **Maior rapidez:** Visto que o ser humano, por exemplo, demora 1 minuto para enviar 10 mensagens, um programa consegue no mesmo tempo enviar milhares de mensagens.
- **Mais trabalho efetuado:** Enquanto o programa está a efetuar ações que antes seriam executadas por profissionais da empresa, neste momento esses podem estar a fazer outras ações que não podem ser automatizadas.

Sendo assim não há dúvidas de que automatizar as tarefas é uma enorme mais valia para qualquer empresa.

Posto isto, serão expostos todos os programas que foram efetuados no decurso do estágio para automatizar tarefas que antes eram feitas manualmente.

6.1 Gestão de endereços de email da empresa

A equipa de marketing envia inúmeros emails para divulgar a empresa e muitos destes não são recebidos. Então chega a uma altura em que a caixa de entrada está cheia de relatórios de erro. Sendo assim é necessário andar constantemente a limpar a caixa de entrada.

Nesse caso tínhamos duas opções:

1. Apagar todas as mensagens de erro recebidas: Assim iríamos perder muitos endereços de email que são importantes para a empresa.
2. Criar um XLS com todos os endereços de e-mail e o respetivo erro.

A melhor solução é sem dúvida a segunda, mas para isso será necessário um programa que percorra toda a caixa de entrada e crie o XLS de forma automática.

O primeiro passo foi baixar as mensagens de erro. O próprio Google através do website '<https://takeout.google.com>' permite, após feito o login na conta que desejamos importar a caixa de entrada, fazer o download da mesma. Esta vem num formato '.mbox'. Este não é o formato perfeito para trabalhar, sendo assim, é necessário abri-lo pelo bloco de notas e guardá-lo num formato '.txt'.

Posto isto apenas temos de analisar o documento '.txt' e identificar os erros. Existem cerca de 120 erros padrão. Os erros podem ser divididos em 3 tipos:

- **Endereço de email será apagado:** Este tipo de erro significa que a conta de utilizador para a qual tentamos contactar deixou de funcionar. Sendo assim estes endereços terão de ser eliminados da nossa base de dados.
- **Endereço de email será alterado manualmente:** Visto que não foi possível enviar o email devido ao endereço de email estar incorreto é necessário manualmente analisar o email, pois a maioria das vezes dá para verificar o erro. Caso não seja detetável a falha é necessário fazer uma rápida pesquisa no google.
- **Endereço de email será alterado automaticamente:** Quando uma empresa possui um novo endereço de email e apresenta uma resposta automática indicando o novo email, este automaticamente altera o endereço.

Alguns exemplos de erros do primeiro tipo são:

- **'Mailbox is full':** O que indica que o endereço de email que tentamos contactar se encontra com a caixa de entrada cheia. Ou seja, deve-se tratar de um endereço que já não se encontra em funcionamento.
- **'This email account doesn't exist anymore':** Significa que o utilizador mudou de correio eletrónico.
- **'The account has been suspended':** O endereço de email foi eliminado.

De seguida serão apresentados alguns erros do segundo tipo:

- **'The account does not exist':** O endereço está errado.
- **'The domain name of the email address is not valid':** O domínio do endereço está incorreto.

Por fim, apresento uns exemplos de erros do terceiro tipo:

- **'Por favor atualize o nosso contacto';**
- **'O email passou a ser';**
- **'Este email irá deixar de existir';**
- **'Este email vai ser desativado';**
- **'Email foi alterado';**
- **'Mudámos de instalações e contactos'.**

Após ter os erros divididos em 3 tipos diferentes o programa irá ler o ficheiro ".txt", lendo email a email, e em cada email deteta o erro.

Quando o programa acaba de ler o ficheiro cria 3 ficheiros correspondentes aos 3 tipos de erro também no formato ".txt".

Estes ficheiros apresentam o 'ID' do erro, a data da mensagem e o endereço de email. No caso do ficheiro referente ao erro do tipo 3 apresenta também o endereço de email a ser alterado.

Por fim o utilizador do programa, ou seja, a equipa de marketing da empresa CentroProduto, terá de ir à base de dados da empresa e apagar/alterar os endereços de email.

6.2 Criar e Importar produtos

A equipa de testes está constantemente a criar produtos para fazer os seus testes. É uma tarefa que mesmo sendo bastante simples despende algum tempo. Visto ser uma ação repetida é algo que pode ser automatizado de forma a poupar bastante tempo.

Sendo assim foi criado um programa de muito fácil utilização. Após executar o programa, este pergunta qual o browser e a instância (predev, dev1, dev2, alpha ou real) onde o utilizador deseja criar os produtos. Posto isto também são pedidos o nome de utilizador e a palavra-chave de forma a que seja possível o programa fazer o login.

Após todas as validações, o programa pede o número de produtos que devem ser criados bem como algumas características:

- Produtos de Compra;
- Produto de Venda;
- Possui Imagem;
- Possui PDF;

Depois é criado um XLS com toda a informação guardada.

Posteriormente de forma automática e à vista dos utilizadores, ou seja, da equipa de testes, o programa abre o browser e a instância desejados, faz login com a conta do utilizador e importa o XLS contendo todos os produtos na conta destes.

Rapidamente o utilizador tem os produtos todos na sua conta.

6.3 LINKEDIN

Além do envio de emails, o LINKEDIN é outro método de marketing da empresa. Este é efetuado através do envio de mensagens a divulgar a empresa a possíveis interessados.

Semelhante à tarefa anterior, é de fácil utilização, mas que por ser uma tarefa tão repetida acaba por ser cansativa e por demorar bastante tempo.

Esta tarefa tem 2 vertentes:

- **Enviar 'Connects':** Estabelece uma ligação entre o utilizador e a pessoa que deseja conectar.
- **Enviar Mensagens:** Permite que o utilizador entre em contacto com outros destinatários através de texto e/ou imagens.

Sendo assim é bastante útil automatizar estas tarefas.

O programa inicia abrindo o LINKEDIN num browser e pedir ao utilizador que faça o seu login e que quando termina pressione 'C' para que o programa continue o procedimento.

De seguida, o utilizador deve pesquisar pessoas/empresas que deseja conectar/enviar mensagem e novamente pressionar 'C' quando terminar.

O programa nesta fase pergunta se o objetivo é enviar 'Connects' ou enviar mensagens e o número de páginas para as quais deseja efetuar essa ação.

Posto isto existem 2 maneiras de concluir a tarefa: enviando de forma imediata os 'Connects'/mensagens para todas as pessoas/empresas que estejam nas páginas escolhidas pelo utilizador ou apresentando o nome de cada pessoa/empresa e deixando o utilizador decidir se deseja apresentar a ação ou se deseja passar para a próxima.

Caso a intenção seja enviar mensagens, é necessário que previamente seja preenchido um XLS que se encontra na página do programa com o nome e texto da mensagem bem como, caso tenha imagem, o nome desta.

Concluindo, caso a ação pretendida seja enviar 'Connects'/Mensagens em série, o programa apresenta melhorias bastante significativas em relação à mesma tarefa feita manualmente. No outro caso, o programa continua a apresentar melhorias, mas menos significativas.

6.4 Descobrir o CAE de empresas

O código CAE é o código que enquadra a atividade económica das empresas e dos profissionais liberais. De acordo com o artigo 151º do CIRS, as atividades exercidas pelos sujeitos passivos de IRS têm que ser classificadas de acordo com a classificação das Atividades Económicas Portuguesas por Ramo de Atividade (CAE), ou de acordo com os códigos mencionados em tabela de atividades aprovada por portaria do Ministro das Finanças.

Este é um atributo importantíssimo no que diz respeito às empresas pois permite-nos saber qual o ramo de atividade da mesma. Mas quando usamos o Web Scraping para recolher informações sobre as mesmas, muitas vezes o CAE não se encontra presente na descrição da empresa.

Sendo assim é necessário manualmente recolher o CAE a diversos websites como 'www.racius.com' ou 'www.gescontact.pt', websites que fornecem informações sobre as empresas como por exemplo: Nome, NIF, morada, capital social, CAE, etc.

Visto tratar-se de uma tarefa repetida, onde apenas se abre um dos websites, procura-se pelo nome da empresa e guarda o CAE, é uma boa oportunidade de criar um programa para automatizar esta tarefa.

Através do programa apenas é necessário preencher um XLS que se encontra na pasta do programa com o nome legal das empresas das quais desejamos descobrir o CAE.

Posto isto, ao iniciar o programa, o utilizador consegue ver o programa a abrir o browser, abrir o website 'www.racius.com', digitar o nome legal da empresa, abrir a informação da mesma, recolher o CAE e introduzi-lo no XLS. Caso a empresa não conste no website, o programa automaticamente abre o outro website 'www.gescontact.pt' e faz o mesmo procedimento. Se a empresa também não constar neste novo website o programa não preenche o CAE e passa para a empresa seguinte.

Num dos testes efetuados pelos utilizadores, foram procurados 185 CAE de empresas diferentes e o programa em menos de 2 minutos conseguiu encontrar e guardar no XLS 157 CAE.

Não é um programa 100% eficaz, mas é uma enorme mais valia nesta tarefa.

6.5 SEND MESSAGES

A equipa de marketing também usa o website da CentroProduto para enviar mensagens a divulgar catálogos feitos pela empresa. Estas mensagens podem ser enviadas em várias línguas, pois o website abriga utilizadores de todas as nacionalidades. Esta é uma ação repetida, onde a equipa de marketing apenas tem de selecionar o destinatário e a mensagem e enviar.

Como nas tarefas anteriores, é uma tarefa de fácil implementação, mas muito demorado o uso de um programa que automatize esta tarefa é fundamental. Sendo assim, foi elaborado um XLS onde em cada linha é possível escolher 1 ou 2 destinatários para receber a mesma mensagem. O XLS possui uma coluna com a língua do destinatário e a respetiva mensagem consoante a língua desejada. Também é possível anexar PDF às mensagens.

O programa permite que o utilizador escolha quantas mensagens deseja enviar seguidas. Normalmente o utilizador escolhe entre 1 a 2 mensagens para ver se a mensagem foi enviada como era pretendido e após o sucedido envia as restantes mensagens de forma contínua.

Além desta funcionalidade também é possível o utilizador escolher o tempo entre o envio de cada mensagem. O tempo sugerido é de 5 segundos para o utilizador do programa conseguir acompanhar o processo do programa.

7 Conclusão

Neste relatório de estágio apresentou-se todo o trabalho desenvolvido em diferentes áreas da empresa CentroProduto. Em cada uma destas áreas foi desenvolvido o trabalho que foi pedido, bem como sugeridas alterações, alguns aceites, outras não.

Na fase de teste ao software da empresa foi proposto o uso de uma plataforma virtual denominada 'Lean Testing' de forma a substituir o uso de XLS. Visto que esta alteração não foi aceite foi decidido então adaptar o XLS usado de forma a obter algo próximo à plataforma sugerida. Estas adaptações consistiram em remover informações duplicadas que existiam nas tabelas do comportamento esperado, criação de uma coluna com o ecrã do erro encontrado, criação de gráficos e alterações da ordem das rotas. Todas estas adaptações foram realizadas. É de salientar que esta foi a fase em que foi despendida a maior parte do tempo do estágio.

Quando estive na área da programação a utilizar a técnica de Web Scraping apenas era usada a biblioteca do Python denominada 'Beautiful Soup'. Este fato condiciona a eficácia da tarefa. Sendo assim foi proposto o uso de uma nova biblioteca denominada 'Selenium' de forma a poder complementar o trabalho. Esta alteração foi o momento mais importante do estágio, pois foi a altura em que uma simples alteração mudou a maneira de pensar da empresa. Esta biblioteca além de ajudar na tarefa de Web Scraping também permite simular ações do utilizador. Sendo assim é possível substituir diversas ações repetidas efetuadas pelos recursos humanos da empresa por simples programas que desenvolvem o mesmo trabalho em muito menos tempo. O uso de funções também foi outra alteração proposta e aceite nesta fase de trabalho.

Na fase de especificação do comportamento esperado foi elaborada uma tabela que conteve 912 ações diferentes. Nesta fase, houve pouco tempo e por isso não surgiu nenhuma sugestão de melhoria. Sendo assim não tive nenhuma sugestão de melhoria.

Com a descoberta do 'Selenium' foi possível a elaboração de diversos programas para automatizar tarefas. Nesta fase foram elaborados 5 programas distintos.

8 Bibliografia

- [1] L. Testing, “Lean Testing,” 12 Março 2018. [Online]. Available: <https://leantesting.com/>. [Acedido em Março 2018].
- [2] P. Pinto, “Programação Python,” 29 Janeiro 2017. [Online]. Available: <https://pplware.sapo.pt/tutoriais/aprenda-dar-uns-toques-na-linguagem-programacao-python/>. [Acedido em Dezembro 2017].
- [3] V. Paruchuri, “Python,” Novembro 2016. [Online]. Available: <https://www.dataquest.io/blog/web-scraping-tutorial-python/>. [Acedido em Dezembro 2017].
- [4] B. Muthukadan, “Location Elements,” Dezembro 2017. [Online]. Available: <http://selenium-python.readthedocs.io/locating-elements.html>. [Acedido em Dezembro 2017].
- [5] R. Renan, “Web Scraping,” Junho 2016. [Online]. Available: <http://www.deviscoming.com/2016/06/voce-sabe-o-que-e-web-scraping.html>. [Acedido em Dezembro 2017].
- [6] P. S. Foundation, “PYTHON,” 2001. [Online]. Available: <https://www.python.org/>.
- [7] I. GIANTOMASO, “Techtudo,” 09 05 2017. [Online]. Available: <http://www.techtudo.com.br/tudo-sobre/gifcam.html>.
- [8] “CentroProduto,” [Online]. Available: <https://www.centroproduto.com/#/pt/>. [Acedido em Maio 2018].
- [9] “Selenium,” [Online]. Available: <https://www.seleniumhq.org/>.